

REMARKS

Claims 1-10, 13-23, and 26-28 are pending and remain.

Rejections under 35 U.S.C. § 103(a) over Schmidt and Chase

5 Claims 1-10, 13-23, and 26-28 stand rejected under 35 U.S.C. § 103(a) as obvious over U.S. Patent No. 6,546,554, to Schmidt et al. ("Schmidt"), in view of U.S. Patent No. 7,240,107 to Chase-Salerno et al. ("Chase"). Applicant traverses.

 The examiner bears the initial burden of factually supporting any *prima facie* conclusion of obviousness, which includes a clear articulation of the reasons or rationale why the claimed invention would have been obvious. MPEP 2142.
10 Exemplary rationales to support a conclusion of obviousness are listed in MPEP 2143, although the list is not all-inclusive.

 The claims appear to be rejected under the rationale outlining combining prior art elements according to known methods to yield predictable results, which includes *inter alia* "a finding that the prior art included each element claimed,
15 although not necessarily in a single prior art reference, with the only difference between the claimed invention and the prior art being the lack of actual combination of the elements in a single prior art reference." MPEP 2143(A). If any of the findings cannot be made, this rationale cannot be used to support a conclusion that the claim would have been obvious. *Id.*

20 Schmidt discloses receiving, installing, and launching applications from a browser on a client computer (Schmidt, Abstract). A JNET helper application is invoked when the browser encounters a metafile with a Java Net Launcher (JNL) file format (Schmidt, Col. 5, line 66-Col. 6, line 8; Col. 7, lines 50-56). The JNET helper application parses the JNL metafile, which is temporarily stored on
25 the client computer, to identify components that are necessary to launch an application specified by the metafile (Schmidt, Col. 7, line 66-Col. 6, line 1). Any necessary components not previously installed on the client computer are identified and downloaded by the JNET helper application (Schmidt, Col. 8, lines 1-24).

30 In contrast, Chase teaches propagating operating system images in a

cluster environment with a multiplicity of nodes (Chase, Abstract). A first node, such as an installation server, is installed with a copy of the operating system and all software required to install a client node (Chase, Col. 2, lines 52-56). The installation server receives an installation request from a client and shuts down the
5 dynamic host configuration protocol ("DHCP") service to eliminate network traffic (Chase, Col. 3, lines 10-17). The client is then installed with a full version of the operating system and any services or images needed to replicate itself (Chase, Col. 3, lines 17-20). Thereafter, the client reboots and is itself a peer server (Chase, Col. 3, lines 20-21).

10 Claim 1 recites a checking mechanism to remotely determine availability of the network service software on the service host system and to verify prerequisites against a runtime environment through the service host system. Claim 14 recites remotely determining by the requesting system, availability of the network service software by the requesting system on the service host system
15 and verifying prerequisites against a runtime environment through the service host system. Claim 28 recites means for remotely determining by the requesting system, availability of the network service software on the service host system and for verifying prerequisites against a runtime environment through the service host system.

20 The Schmidt-Chase combination fails to teach such limitations. Rather, Schmidt teaches a client computer that includes one or more Java Runtime Environments (JREs) and a helper application (Schmidt, Col. 5, line 55-Col. 6, line 3). When a browser on the client computer encounters a metafile on a server with a new Java Net Launcher (JNL) file format, a copy of the JNL metafile is
25 downloaded to a local temporary file on the client computer (Schmidt, Col. 7, lines 59-65). The helper application is invoked to select an appropriate JRE based on the JNL metafile (Schmidt, Col. 6, lines 3-14; Col. 7, lines 50-65). If the appropriate JRE has not been installed on the client, the helper application automatically identifies and installs the appropriate JRE (Schmidt, Col. 6, lines
30 14-17; Col. 8, lines 1-13). The helper application parses the temporary JNL metafile to identify necessary components for the metafile (Schmidt, Col. 7, line

66-Col. 8, line 1). Once identified, the helper application downloads and installs the necessary components, if any, on the client computer (Schmidt, Col. 8, lines 1-5). Thus, in Schmidt, the metafile located on the server merely provides information *for use by* the helper application to identify necessary components, instead of testing a requesting system to verify prerequisites (Application, page 7, line 24-page 8, line 31; FIGURE 3). Therefore, Schmidt teaches a passive metafile with information *for use by* a helper application, rather than to verify prerequisites against a runtime environment through the service host system, per Claims 1, 14, and 28. Chase fails to remedy the shortcomings of Schmidt.

The prior art made of record and not relied upon has been reviewed by the applicant and is considered to be no more pertinent than the prior art references already applied.

Claims 1-10, 13-23, and 26-28 are believed to be in condition for allowance. Entry of the foregoing amendments is respectfully requested.

Reconsideration of the claims, withdrawal of the finality of the Office action, and a Notice of Allowance are earnestly solicited. Please contact the undersigned at (206) 381-3900 regarding any questions or concerns associated with the present matter.

Respectfully submitted,

Dated: December 24, 2009

By:



Krista A. Wittman, Esq.
Reg. No. 59,594

Cascadia Intellectual Property
500 Union Street, Suite 1005
Seattle, WA 98101

Telephone: (206) 381-3900
Facsimile: (206) 381-3999